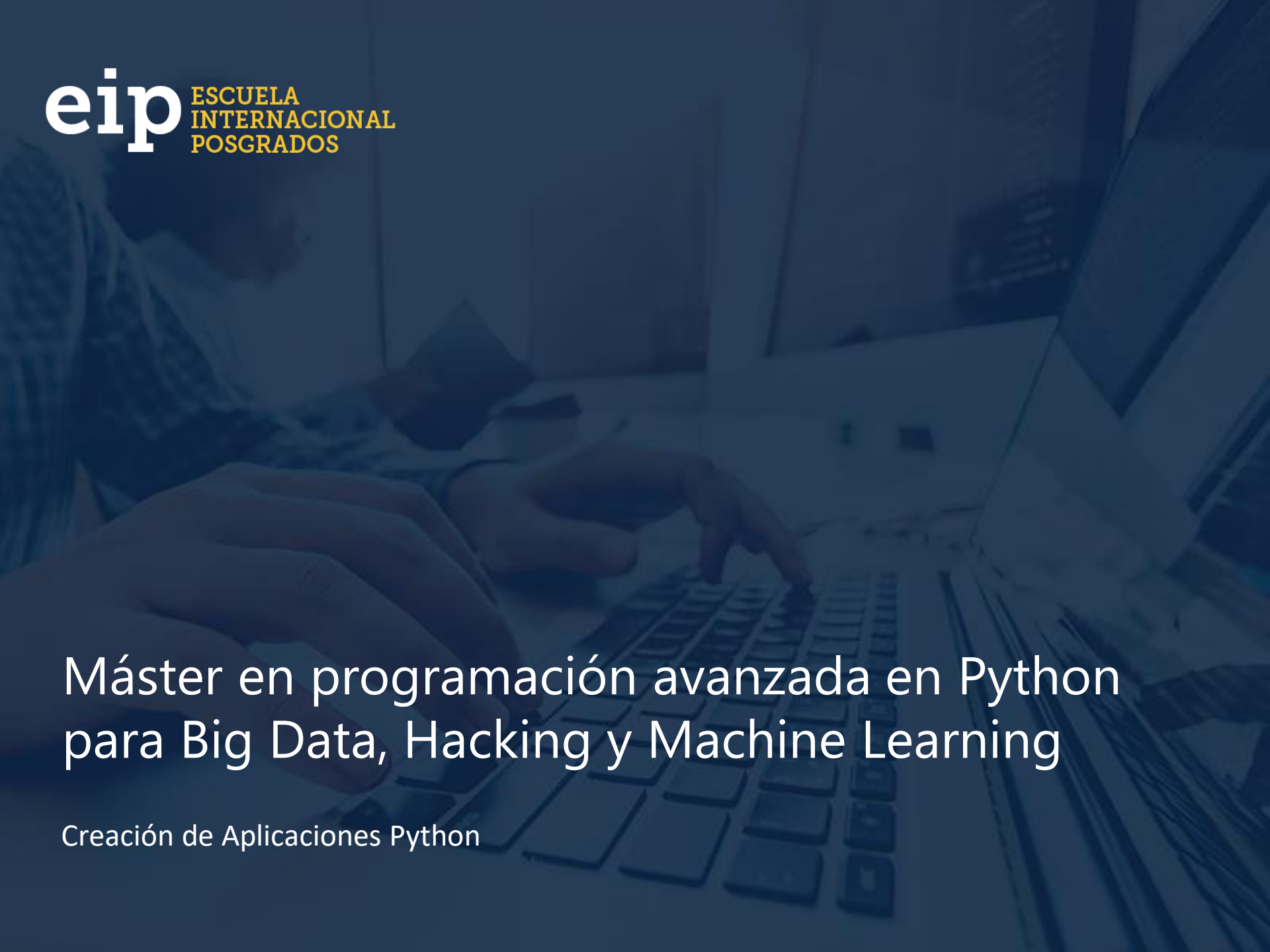

ESCUELA INTERNACIONAL POSGRADOS

Máster en programación avanzada en Python para Big Data, Hacking y Machine Learning

Creación de Aplicaciones Python

LECCIÓN 08

Flask Framework - API REST

ÍNDICE

INTRODUCCIÓN

En esta lección aprenderemos a realizar un API REST usando Flask como Framework de Python, realizando los métodos HTTP más importantes que son GET, POST, PUT y DELETE.

OBJETIVOS

Al finalizar esta lección serás capaz de:

1. **Aprender a crear una API REST usando el framework Flask**

1. Crear la aplicación en Flask

1) Instalar la librería flask: pip install flask

2) Importar flask en el archivo __init__.py: from flask import Flask

3) Crear la aplicación con nombre “app”: app = Flask(__name__)

4) Ejecutar la aplicación en:

```
if __name__ == '__main__':
        app.run(debug=True)
```

5) Definir la URL y el método de la aplicación:

```
@app.route('/', methods=['GET'])
def home():
        return " "
```

2. Método POST

Para ello ponemos:

@app.route("/insertData/", methods=["POST"]

```python
# Ruta de inicio "/insertData/", metodo GET
@app.route('/insertData/', methods=['POST'])
def insertdata():
    # definir 'data' como el conjunto de datos
    # que se reciben a través del Postman:
    data = request.data
    data = json.loads(data)
    # insertar dato en csv:
    with open('iris.csv', 'a', newline='') as csvfile:
        fieldnames = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writerow({'sepal_length': data['sepal_length'],
                         'sepal_width': data['sepal_width'],
                         'petal_length': data['petal_length'],
                         'petal_width': data['petal_width'],
                         'species': data['species']})

        print("writing complete")
    return data
```

3. Método PUT

Para ello ponemos:

@app.route("/updateData/", methods=["PUT"]

```python
# Ruta de inicio "/updateData/", metodo PUT
@app.route('/updateData/', methods=['PUT'])
def updatedata():
    # definir 'data' como el conjunto de datos
    # que se reciben a través del Postman:
    data = request.data
    data = json.loads(data)
    df = pd.read_csv('iris.csv')

    # sustituimos la última fila del dataset cada uno de los valores
    # con los datos que recibimos 'data':
    df.loc[df.index[-1], 'sepal_length'] = data['sepal_length']
    df.loc[df.index[-1], 'sepal_width'] = data['sepal_width']
    df.loc[df.index[-1], 'petal_length'] = data['petal_length']
    df.loc[df.index[-1], 'petal_width'] = data['petal_width']
    df.loc[df.index[-1], 'species'] = data['species']
    # convertir a csv
    df.to_csv('iris.csv', index=False)
    # mostrar el último dato en formato Json:
    result = df.iloc[-1].to_json(orient="index")
    return result
```

4. Método DELETE

Para ello ponemos:

@app.route(“/deleteData/”, methods=[“DELETE”]

```python
# Ruta de inicio "/deleteData/", metodo DELETE
@app.route('/deleteData/', methods=['DELETE'])
def deleteData():
    df = pd.read_csv('iris.csv')
    # Eliminar la última fila
    df.drop(df.index[-1], inplace=True)
    # convertir a csv
    df.to_csv('iris.csv', index=False)
    # mostrar el último dato en formato Json:
    result = df.iloc[-1].to_json(orient="index")
    return result
```

MUCHAS GRACIAS POR SU ATENCIÓN

eip ESCUELA INTERNACIONAL POSGRADOS

jmpena@grupomainjobs.com
imaniega@grupomainjobs.com

José Manuel Peña:
linkedin.com/in/josé-manuel-peña-castro-7566b349
Isabel Maniega:
linkedin.com/in/isabel-maniega-cuadrado-40a8356b

twitter.com/eiposgrados

facebook.com/eiposgrados

instagram.com/eiposgrados